Applied Energy 87 (2010) 1782–1792

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Metamodel-assisted evolutionary algorithms for the unit commitment problem with probabilistic outages

Chariklia A. Georgopoulou, Kyriakos C. Giannakoglou *

National Technical University of Athens, School of Mechanical Engineering, Lab. of Thermal Turbomachines, Parallel CFD and Optimization Unit, P.O. Box 64069, Athens 157 10, Greece

ARTICLE INFO

Article history:
Received 14 April 2009
Received in revised form 16 October 2009
Accepted 18 October 2009
Available online 18 November 2009

Keywords:
Unit commitment
Probabilistic outages
Metamodels
Two-level evolutionary algorithms

ABSTRACT

An efficient method for solving power generating unit commitment (*UC*) problems with probabilistic unit outages is proposed. It is based on a two-level evolutionary algorithm (*EA*) minimizing the expected total operating cost (*TOC*) of a system of power generating units over a scheduling period, with known failure and repair rates of each unit. To compute the cost function value of each *EA* population member, namely a candidate *UC* schedule, a Monte Carlo simulation must be carried out. Some thousands of replicates are generated according to the units' outage and repair rates and the corresponding probabilities. Each replicate is represented by a series of randomly generated availability and unavailability periods of time for each unit and the *UC* schedule under consideration accordingly. The expected *TOC* is the average of the *TOCs* of all Monte Carlo replicates. Therefore, the *CPU* cost per Monte Carlo evaluation increases noticeably and so does the *CPU* cost of running the *EA*. To reduce it, the use of a metamodel-assisted *EA* (*MAEA*) with on-line trained surrogate evaluation models or metamodels (namely, radial-basis function networks) is proposed. A novelty of this method is that the metamodels are trained on a few "representative" unit outage scenarios selected among the Monte Carlo replicates generated once during the optimization and, then, used to predict the expected *TOC*. Based on this low cost, approximate pre-evaluation, only a few top individuals within each generation undergo Monte Carlo simulations. The proposed *MAEA* is demonstrated on test problems and shown to drastically reduce the *CPU* cost, compared to *EAs* which are exclusively based on Monte Carlo simulations.

1. Introduction

Given a power demand distribution over a scheduling period of *T* hours and a power generating system of *M* units, the associated *UC* problem is solved by taking into account probabilistic unit outages, determined by their failure and repair rates. To solve this problem, an optimization algorithm and a software for the evaluation of commitment schedules subject to (un)availability constraints are necessary. It is also important to find the optimal solution at affordable *CPU* cost; this is where this paper is focusing on.

A literature survey reveals different *UC* models and solution methods [1–10] accounting for: (a) deterministic or stochastically varying power demand distributions over time, (b) time constraints related to the up, down, etc., times of units, (c) various costing models including fuel, start-up or shut-down and maintenance costs and/or, (d) uncertainties of the availability of units due to probabilistic outages. Concerning the last point, Monte Carlo simulations are appropriate for evaluating candidate solutions to *UC* problems with probabilistic outages. A Monte Carlo simulation requires the knowledge of the failure and repair rates of each unit and sets up an adequate number (*J*) of replicates corresponding to randomly selected unit availability/unavailability scenarios. Each candidate solution (*UC* schedule) to be evaluated must be adapted to these replicates. Then, *J* evaluations are performed to compute their *TOCs* and, through averaging, the expected total operating cost to be minimized. A single evaluation based on a Monte Carlo simulation costs about *J* times the cost of evaluating a single *UC* schedule with all units available. Therefore, an *EA* based optimization becomes computationally expensive and methods for reducing its *CPU* cost are to be devised.

In [11], a two-level optimization technique based on *EAs* was proposed for solving *UC* problems without probabilistic unit outages. To account for stochastic power demand variations, an issue which is beyond the scope of the present paper, a two-objective optimization problem was formulated and solved. The method proposed in [11] splits the search for the optimal solution in two communicating levels. The low level explores the decision space by solving a coarser problem at low *CPU* cost, whereas the high level solves the fine problem faster by exploiting promising solutions from the low level. Techniques for coarsening the *UC* problem so as

* Corresponding author. Tel.: +30 210 7721636.
E-mail address: kgianna@central.ntua.gr (K.C. Giannakoglou).

0306-2619/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.apenergy.2009.10.013

to handle a smaller number of decision variables on the low level and expanding the coarse problem best solutions which migrate to the high level were necessary. Additional assistance to the fine level search was offered by partitioning the scheduling horizon into subperiods where *UC* problems of smaller size are sequentially/iteratively solved via *EAs*. As shown in [11], the two-level method performs much faster than conventional *EAs*.

The *CPU* cost of the method proposed in [11] is low, due to its two-level structure and the fragmentation of the scheduling period into subperiods where semi-isolated *EAs* are employed. However, introducing a Monte Carlo based evaluation per candidate solution (with $J \approx 10^4$ replicates per evaluation, see below), for the needs of the present paper, increases the overall *CPU* cost by three to four orders of magnitude. To maintain the advantages of the *EA*-based optimization method while extending it to cope with probabilistic unit outages with just a reasonable *CPU* cost overhead, the use of surrogate evaluation models is proposed.

Assisting *EA*-based optimization methods for computationally demanding problem by surrogate models (the so-called metamodels) is not new. The metamodels displace many calls to the costly evaluation software during the evolution, leading to inexpensive search methods, [12,13,17]. A very efficient metamodel-assisted *EA* (*MAEA*) was proposed in [14–17] for complex aerodynamic optimization problems. The metamodels (radial-basis function, *RBF*, [20], networks, trained on sets of real numbers) were used along with the problem-specific evaluation model, in an interleaving way during the evolution. Metamodel-based pre-evaluation served to screen out non-promising population members and restrict costly evaluations only to a few most promising among them. For each new candidate solution, a different "local" metamodel was trained on a small number of previously evaluated and archived neighboring individuals and used to approximate its cost function value(s).

In a *UC* problem, where the design variables are binary digits denoting *ON* and *OFF* unit states, it is absolutely ineffective to use metamodels in the aforementioned way. So, in the *MAEA* for *UC* problems with probabilistic unit outages, a new way of training and using metamodels is proposed. Since the target is to compute the expected *TOC* resulting from J (some thousands of) randomly generated failure–repair scenarios, the metamodel is presented with the computed *TOC* values of a few ($J_{ind} \ll J$, say for instance $J_{ind} \approx 10$) preselected unit availability scenarios and approximates the expected *TOC*. So, the pre-evaluation of each population member, in each generation, requires: (a) the evaluation of this *UC* schedule adapted to J_{ind} preselected units failure/repair scenarios, (b) the training of a "local" *RBF* network to be presented with the previously computed *TOCs* paired with the corresponding expected *TOCs* for the J_{ind} scenarios, for approximating the expected *TOC* and (c) just a few costly Monte Carlo simulations for re-evaluating the most promising among the population members.

The new method is demonstrated on two test problems and shown to constantly and noticeably reduce the *CPU* cost compared to conventional *EAs*.

2. *UC* problem formulation with probabilistic unit outages

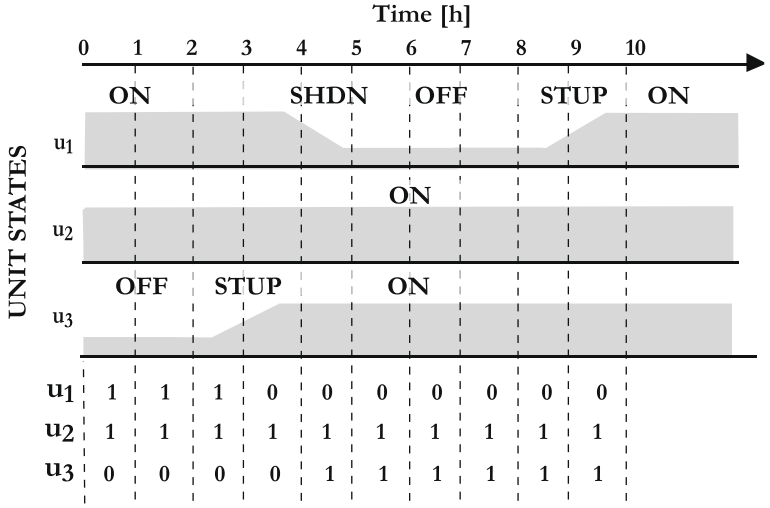
A system with M units should satisfy a power demand distribution ($d^{(j)}$, $1 \leqslant j \leqslant T$) over the scheduling period of T hours. After solving the *UC* problem (either without or with stochastic unit outages), each unit (subscript i), at each hour (superscript j), is assigned a binary digit $\left(s_i^{(j)} = 1 \text{ or } 0\right)$ representing *generating* (*ON*) or *non-generating* (*OFF*, start-up: *STUP* and shut-down: *SHDN*) states, respectively. Such a binary coding, see Fig. 1, fits perfectly to the *EAs* but since the chromosome consists of *MT* binary digits, a conventional *EA* becomes highly *CPU* demanding for high M and/or T values. Note that $s_i^{(j)} = 0$ corresponds to any of the three *non-generating* states (*OFF*, *STUP* or *SHDN*) which can be distinguished by taking into account the previous or next states of the same unit.

With known $s_i^{(j)}$ $(1 \leqslant i \leqslant M)$, the optimal loads $x_i^{(j)}$ of the generating units ($\frac{P_{min,i}}{P_{max,i}} \leqslant x_i^{(j)} \leqslant 1$, where $P_{min,i}$ and $P_{max,i}$ are the minimum and maximum power production capacities) result from the minimization of the hourly cost. To this end, T independent hourly economic dispatch problems are solved using the augmented Lagrange multipliers method (*ALM*, [18]).

In the present method, *TOC* (in MW h or $) sums up fuel consumption and penalties for unmet demands, as follows:

$$TOC = \sum_{j=1}^{T} \sum_{i=1}^{M} OC_i^{(j)} + \sum_{j=1}^{T} \Phi(\Delta d^{(j)}) \tag{1}$$

in general, the hourly operating cost $OC_i^{(j)}$ is given by quadratic polynomials with known coefficients (a_i, b_i, c_i) and also includes *STUP* and *SHDN* costs $\left(C_i^{STUP},\ C_i^{SHDN}\right)$,



EA chromosome: 1101101100100110110110110110 11

Fig. 1. Binary coding of a *UC* schedule, for $M = 3$ units and $T = 10$ h. At each hour, the state of each unit is assigned a binary digit: $s_i^{(j)} = 1$ or 0. Note that $s_i^{(j)} = 0$ represents *non-generating* states, i.e. *OFF*, *STUP* and *SHDN* states. The three states can readily be distinguished by examining previous $\left(s_i^{j-1}\right)$ and next $\left(s_i^{j+1}\right)$ bits and states. $s_i^{(j)} = 1$ stands for the *generating* state (*ON*).